

Paper 224-2009

## BUILDING DRILL-DOWN CAPABILITIES DYNAMICALLY IN SAS® OUTPUT USING BASE SAS®

Hong Zhang, Mathematica Policy Research, Inc., Princeton, NJ  
Ronald Palanca, Mathematica Policy Research, Inc., Princeton, NJ  
Mark Beardsley, Mathematica Policy Research, Inc., Princeton, NJ  
Barbara Kolln, Mathematica Policy Research, Inc., Princeton, NJ

### ABSTRACT

*“Can you give me a list of ID numbers where ....?”* If you are a SAS® programmer, requests of this type might sound very familiar to you. Such requests occur primarily when research staff members determine that data values may be outside an acceptable range, are inconsistent, or that the data for certain cases should be examined more closely. Wouldn't it be nice if they could look up the ID numbers on their own? This paper presents a tool for enabling them to do just that. The application dynamically creates drill-down links in SAS output, enabling research staff to view listings of ID numbers behind each data value of any variable.

Creating one or more hyperlinks in fixed positions can be accomplished fairly easily in some types of SAS output. However, creating hyperlinks dynamically in every data cell of a report is much more difficult, particularly when the source SAS dataset changes over time. The challenges in developing the application included (1) identifying the positions of data cells in the SAS output, (2) dynamically creating separate HTML listings of ID numbers for each data value, (3) matching data cells correctly with their corresponding HTML listings, and (4) adding hyperlinks to each data cell on the fly. The application uses SASHELP.VCOLUMN to dynamically identify the input dataset and variables, as well as PROC REPORT'S CALL DEFINE facility to establish hyperlinks in data cells. A series of macros automates the process.

### INTRODUCTION

When researchers wish to investigate or verify data values for selected cases (observations), they often ask programmers to provide the ID numbers of those cases. An application that allows non-programmers to view listings of ID numbers associated with responses of interest could be a valuable tool in the data examination and verification process. The benefits include (1) a more efficient use of staff time, which allows programmers to spend additional time on more complex programming tasks; and (2) a reduction in the overall costs of a research project.

An application that allows staff to identify specific cases in a dataset should have dynamic capabilities; that is, it should be able to reflect the changing nature of the data as more cases are included and as responses show increasing variability. The application described in this paper creates either one-way or two-way (crosstab) frequency output in HTML format with hyperlinks, enabling users to connect (“drill down”) to lists of respondent ID numbers or to simple reports that present other variables in addition to ID numbers. The program was designed to allow researchers to select variables from any SAS dataset and to review their frequencies, either in one-way or crosstab output. The HTML listings can be saved to a file or printed out.

The main programmatic challenges in developing the application using Base SAS were (1) Identifying the positions of data cells in the frequency output, (2) matching data cells (frequency counts) with their corresponding HTML listings, and (3) adding hyperlinks to each data cell. Output produced by the program is generated dynamically; that is, it reflects the current range of values of the selected variables, which in many scenarios can change significantly over the course of data collection.

### DATA

The application is described using a SAS dataset (N=360), called STUDENT, which includes ID, YEAR (year of birth), GRADE, GENDER, AGE, SCHOOLID, and STATE. GRADE and YEAR were chosen to illustrate both one-way and two-way frequency output. Figure 1 presents sample records from the STUDENT dataset.

Figure 1. Student Sample Records

ID	YEAR	GRADE	GENDER	AGE	SCHOOLID	STATE
4111723	M	G1	M	8	17	VA
4114134	2001	G1	F	8	13	NJ
4115134	1999	G3	F	10	59	FL
4115189	1999	G3	M	10	76	VA
4115289	.	G3	M	8	35	CA
4115723	2001	G1	M	8	20	NY
4115778	2001	G1	F	8	94	CA

## INTERACTIVE SET UP

The program can be run to obtain either one-way frequencies or crosstabs. %Window was added to make the selection interactive (see Figure 2).

Figure 2. Set-Up Screen

```
Command ===>

Building Drilldown Capabilities Dynamically in SAS Output using Base SAS

Are we doing Crosstab or One way Frequency?(1-One way Freq, 2-Crosstab)  _

Press ENTER to continue
```

After the choice between one-way or crosstab output is made, a second screen allows users to type in the names of variables (in an existing SAS dataset) whose values they wish to investigate, or to select all of the variables.

## DRILLING DOWN IN ONE-WAY FREQUENCIES SELECTING VARIABLES AND CREATING HTML REPORTS FOR DATA VALUES

The program permits a choice between one-way frequencies on all variables in a dataset or on a subset of variables. %LET statements are used to create the name of the SAS dataset and its location as macro variables. Three macros are then executed in this part of the program.

The first macro, %CREATE\_TABLES, runs individual frequencies on the variables selected in the %Window screen. Each set of frequency output is then saved as an individual SAS dataset, which is named according to the variable name. Each individual frequency count will have a number, called UID, assigned to it. The UID will be equal to \_n\_, the internal SAS sequential number for each observation in a dataset. The combination of VARNAME and UID in each dataset is unique. All individual frequency output datasets are then combined into one dataset, called ALL\_FREQS (see Figure 3). The %CREATE\_TABLES macro is executed dynamically using parameters passed to it from the next macro, %TABLE\_DRIVER.

```
%macro create_tables (lib, dataset, col, type);
proc freq data=&lib.&dataset;
  table &col./list missing out=&col. outcum;
run;
data &col.;
  set &col.;
  length value $300. varname $20. vartype $4.;
  varname="&col.";
  uid = _n_;
  value=STRIP(put(&col.,8.));
```

```

vartype = "&type";
drop &col.;
run;
data All_Freqs;
set All_Freqs &col.;
run;
%mend create_tables;

```

The second macro, %TABLE\_DRIVER, determines if frequencies are to be run on all variables or only on selected variables (based on the entry made in the interactive portion of the program). The entry is passed as a macro variable to the SAS program, and is used in the %TABLE\_DRIVER macro. %TABLE\_DRIVER creates parameters dynamically and passes them to the %CREATE\_TABLES macro using CALL EXECUTE.

```

%macro table_driver;
data _null_;
set sashelp.Vcolumn;
%if &num_fields.^=1 %then %do;
if libname = 'DSN' and memname=upcase("&dataset");
call execute("%create_tables('||libname||','||memname||','||name||','||type||')");
%end;
%else %do;
if libname = 'DSN' and memname=upcase("&dataset") and upcase(name) in
(&selected_fields);
call execute("%create_tables('||libname||','||memname||','||name||','||type||')");
%end;
%mend table_driver;

```

Figure 3. All\_Freqs Dataset

COUNT	CUM_FREQ	value	varname	vartype	uid
1	1	.	YEAR	num	1
1	2	M	YEAR	num	2
45	47	1996	YEAR	num	3
39	86	1997	YEAR	num	4
35	121	1998	YEAR	num	5
111	232	1999	YEAR	num	6
44	276	2000	YEAR	num	7
64	340	2001	YEAR	num	8
20	360	2002	YEAR	num	9
52	52	G1	GRADE	char	1
61	113	G2	GRADE	char	2
68	181	G3	GRADE	char	3
65	246	G4	GRADE	char	4
57	303	G5	GRADE	char	5
57	360	G6	GRADE	char	6

The third macro, %CREATE\_HTMLS, creates a series of individual datasets. Each dataset is a subset of the STUDENT dataset and represents a unique combination of the variables VARNAME and UID in the ALL\_FREQS dataset (see Figure 3). ODS HTML and PROC PRINT are then used to create HTML pages for each table. Each HTML page is named based on a unique combination of VARNAME and UID. For example, if the variable VALUE in the ALL\_FREQS dataset is equal to 1999, UID would be 6. A dataset will then be created using the filter, YEAR = 1999, and saved as an HTML file named YEAR\_6.html. Once again, CALL EXECUTE is used to dynamically create the macro variables that are passed to the %CREATE\_HTMLS macro.

```

%macro create_htmls (var,value,uid,type);
ods html file="&path.\&var._&uid..html"; *variable & row number combination*;
proc print data=&var._&uid. noobs;
where &var.="&value.";
var ID, Gender, Age, SchoolID, State
title "Variable: &var., Value:&value.";
%mend create_htmls;
data _null_;
set All_Freqs;
call execute ("%create_htmls('||STRIP(var)||','|| STRIP(value)||','|| STRIP (uid)||','||STRIP(vartype)||')");
run;

```

## CREATING THE FREQUENCY REPORT WITH HYPERLINKS

Each frequency output dataset created by the %CREATE\_TABLE macro will then be displayed in a report using PROC REPORT. The %CREATE\_REPORT macro is executed dynamically by the %RPT\_DRIVER macro. %RPT\_DRIVER is similar to the %TABLE\_DRIVER macro in that it takes into account only the variables that were selected in %Window for one-way frequencies. Variable names are then passed as parameters from the %RPT\_DRIVER macro to the %CREATE\_TABLE macro.

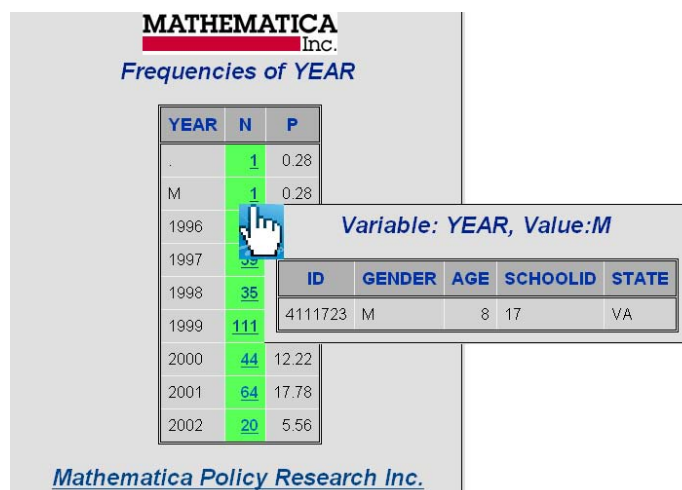
In the COMPUTE BLOCK for the column *Count* (N) in the PROC REPORT code below, a URLSTRING is created based on the variable name passed as a parameter and the UID from the input frequency dataset. For example, if the variable name passed is YEAR and the UID is 1 from the YEAR frequency dataset, then a URLSTRING will be created with the name of YEAR\_1.html. If the URL address just created is one of the HTML files produced by the %CREATE\_HTML macro, then a hyperlink will be created between the two files using the CALL DEFINE statement. The macro variable &FIELD and the variable UID are used to identify the row of the data cell in the report. For each row, the facility CALL DEFINE is employed to identify the columns and setup the URL and style for each column in that specific row. Formatting of columns and cells is done inside the COMPUTE block.

```
%macro create_report (field,lbl);
  proc report data=&field. headskip headline ls=256 nowd missing;
    column uid value COUNT PERCENT;
    define uid/display noprint;
    define value/display "&field." width=255 flow;
    define COUNT / 'N';
    define PERCENT / 'P' f=7.2;
    title2 "Frequencies of &field.";
    compute COUNT;
    urlstring="&path.\&field._|| STRIP(uid)||".html";
    if fileexist (urlstring) then do;
      call define ("_c3_", "URL", urlstring);
      call define ("_c3_", "style", "style=[background=light greenfont_ weight = bold]");
    end;
  endcomp;
run;
%mend create_report;

%macro rpt_driver;
  %if &VARS.^=2 %then %do;
    data _null_;
    set sashelp.Vcolumn;
    if libname='DSN' and memname=upcase("&dataset");
    call execute ("%create_report('||name||',%str('||label||')");
    run;
  %end;
  %else %do;
    data _null_;
    set sashelp.Vcolumn;
    if libname ='DSN' and memname=upcase("&dataset") and upcase(name) in
    (&selected_fields) ;
    call execute ("%create_report('||name||',%str('||label||')");
    run;
  %end;
%mend rpt_driver;
```

As noted previously, the program uses the variables YEAR and GRADE from the STUDENT dataset for illustrative purposes. Figure 4 shows how the one-way frequency output works. By clicking on the frequency (N) for the value "M" (missing) in the frequency report, the user obtains a drill-down listing of ID numbers that have missing values for YEAR (in this example, YEAR is missing for only one case). In addition to ID, the listing of cases also includes GENDER, AGE, SCHOOLID, and STATE. All of the listings of individual cases have the name of the variable and its value in the title.

Figure 4. One-Way Frequency Output



## DRILLING DOWN IN TWO-WAY FREQUENCIES MACRO ASSIGNMENT AND CREATING UNIQUE VALUES

To create a two-way frequency with drill-down features, the set of unique values for each variable must be identified. In the example presented, the unique values of YEAR, the row variable, are captured in &VAR1 and the unique values of GRADE, the column variable, are in &VAR2.

```
%let var1=Year ;
%let var2=Grade;

proc freq data=student noprint;
  table &var1.*&var2./ out=outdata missing;
run;

proc sql;
  create table &var2 as
  select distinct &var2 from outdata;
  create table &var1 as
  select distinct &var1 from outdata;
quit;
```

## CREATING THE DRILL-DOWN DATA

Each cell in the crosstab of YEAR and GRADE will be linked to a data file that will provide more information about the students identified by the unique combination of YEAR and GRADE. Creating the data is straightforward; creating a unique name for the data file for each hyperlink is more complicated. The program creates separate datasets for each variable in the crosstab and assigns a unique number for each row (&N\_). ROW is the name of the variable created for the unique number in the &VAR1 dataset. For the &VAR2 dataset, the variable name is COL.

```
data &var1;          data &var2;
  set &var1;        set &var end=i;
  row = &N_;       col = &N_;
run;               run;
```

Each dataset is merged with the crosstab dataset, called OUTDATA, to obtain the unique combinations of &VAR1 and &VAR2. This dataset is called FINAL and is then merged with the STUDENT dataset (Figure 5). Each combination of the ROW and COL variables is equivalent to a combination of YEAR and GRADE.

Figure 5. Two-Way Frequency Output

ID	YEAR	GRADE	col	row
4111723	2001	G1	1	6
4114134	2001	G1	1	6
4115134	1999	G4	4	4
4115189	1999	G3	3	4
4115289	2001	G1	1	6
4115723	2001	G1	1	6

A macro called %CREATE\_EACH\_HTML outputs multiple HTML files, which represent combinations of the ROW and COL variables. The macro is executed in the DATA NULL step for each unique combination of &VAR1 (YEAR, the row variable) and &VAR2 (GRADE, the column variable).

```
%macro create_each_html(row,col);
  ODS html file="&PROJECT._&row._&col..html" STYLE = barrettsblue;
  proc print data=FINAL noobs;
    where row=&row and col=&col;
    var ID GENDER AGE SCHOOLID STATE;
    title "<IMG SRC = 'MPRlogo.jpg'>";
    title2 "&var1.=&vvar1, &var2.=&vvar2" ;
  run;
%mend create_each_html;

data _null_;
  set Final;
  call execute("%create_each_html('|| STRIP (row)||','|| STRIP(col)||');");
run;
```

### CREATING THE FREQUENCY REPORT WITH HYPERLINKS

The two-way frequency is output using PROC REPORT. Using a DO loop in the COMPUTE statement for the subcolumn COUNT, each cell in the COUNT field is assigned a URL address. ROW = 1 and COL = 1 would be 1\_1.html and so on. If the value of the cell is greater than zero and the corresponding HTML file is one of those created in %CREATE\_HTML, then the cell is linked to the HTML file using the CALL DEFINE statement. Formatting of the columns and cells is added in the Compute statement.

```
ODS html file="crosstab_&PROJECT..html" STYLE = barrettsblue ;
proc report data=outdata headskip headline ls=255 nowd missing;
  column row &var1. ("&var2" &var2. (COUNT PERCENT));
  define &var1./group "&var1" STYLE=[BACKGROUND= barrettsblue];
  define row/group noprint;
  define &var2./across ' ' order=INTERNAL;
  define COUNT /'N';
  define PERCENT / 'P' f=7.2 ;
  title "<IMG SRC = 'MPRlogo.jpg'>";
  title2 "Table of &var1 by &var2";
  compute count;
  %do l=1 %to %eval(&varnum);
    urlstring_&l.="&PROJECT._"|| STRIP (row)||"_"|| STRIP(&l.)||".html";
    if fileexist(urlstring_&l.) and _c%eval(2*&l.+1)_>0 then do;
      call define("_c%eval(2*&l.+1)_" , "URL", urlstring_&l. );
      call define(_COL_, "style", "STYLE=[FONT_WEIGHT = BOLD]");
    end;
  %end;
endcomp;
compute percent;
calldefine(_COL_, "style", "STYLE=[BACKGROUND=YELLOW]");
endcomp;
run;
```

As shown in Figure 6 below, if the cell is selected (N=3) for YEAR = 2002 and GRADE = G2, the inset HTML file is presented, which provides more detailed information about these three cases.

Figure 6. Two-Way Frequency Report with Hyperlinks

<b>MATHEMATICA</b> Inc. Table of YEAR by GRADE															
		GRADE													
		G1		G2		G3		G4		G5		G6			
YEAR	N	P	N	P	N	P	N	P	N	P	N	P	N	P	
.	<a href="#">1</a>	0.28	.	.	.	.	.	.	.	.	.	.	.	.	
M	.	.	<a href="#">1</a>	0.28	.	.	.	.	.	.	.	.	.	.	
1996	.	.	.	.	.	.	.	<a href="#">2</a>	0.56	<a href="#">3</a>	0.83	<a href="#">40</a>	11.11	.	
1997	.	.	.	.	.	.	<a href="#">2</a>	0.56	.	<a href="#">22</a>	6.11	<a href="#">15</a>	4.17	.	
1998	.	.	.	.	.	.	.	<a href="#">4</a>	1.11	<a href="#">29</a>	8.06	<a href="#">2</a>	0.56	.	
1999	<a href="#">1</a>	0.28	.	.	.	.	<a href="#">53</a>	14.72	<a href="#">54</a>	15.00	<a href="#">3</a>	0.83	.	.	
2000	.	.	.	.	<a href="#">34</a>	9.44	<a href="#">8</a>	2.22	<a href="#">2</a>	0.56	.	.	.	.	
2001	.	.	<a href="#">35</a>	9.72	<a href="#">24</a>	6.67	.	.	.	.	.	.	.	.	
2002	.	.	<a href="#">16</a>	4.44	<a href="#">3</a>	0.83	.	.	.	.	.	.	.	.	

ID	GENDER	AGE	SCHOOLID	STATE
10201333	M	7	69	FL
10201999	M	7	57	VA
17652289	M	7	66	VA

**CONCLUSION**

The application presented in this paper can be helpful to both SAS programmers and non-programmers in situations where data values suggest a need for further investigation. This application requires only Base SAS and can be used with version 8.2 or higher.

**CONTACT INFORMATION**

Please feel free to contact the authors:

- Hong Zhang - [HZhang@mathematica-mpr.com](mailto:HZhang@mathematica-mpr.com)
- Ronald Palanca - [RPalanca@mathematica-mpr.com](mailto:RPalanca@mathematica-mpr.com)
- Mark Beardsley - [MBeardsley@mathematica-mpr.com](mailto:MBeardsley@mathematica-mpr.com)
- Barbara Kolln - [BKolln@mathematica-mpr.com](mailto:BKolln@mathematica-mpr.com)

Mathematica Policy Research, Inc.  
 600 Alexander Park  
 Princeton, NJ 08543  
 (609) 799-3535

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

□□□□

Other brand and product names are registered trademarks or trademarks of their respective companies.□□

**APPENDIX: SAS PROGRAM**

```

options NOXSYNC NOXWAIT xmin MPRINT SYMBOLGEN ls=200 ps=50;

/* Optional project name */
%let PROJECT=SUGI2009;
/* SAS input file */
%let dataset=STUDENT;
/* Location where SAS input file resides */
%let path=c:\temp;
libname dsn ".";

/* Delete existing HTML files */
%MACRO RemoveFiles;
  x del "&path\html\*.html";
%MEND;
%RemoveFiles;

%window start
#4 @22 'Building Drilldown Capabilities Dynamically in SAS Output using Base SAS' COLOR = BLUE attr =
underline
#9 @10 'Are we doing Crosstab or One way Frequency?(1-One way Freq, 2-Crosstab)' COLOR = RED @90 FRQ 2
attr=underline
color=RED REQUIRED = YES
14 @10 'Press ENTER to continue' COLOR = RED;
%display start;

%MACRO FREQ;
/* one way frequencies */
%IF &FRQ=1 %THEN %DO;
%window start
#4 @22 'Building Drilldown Capabilities Dynamically in SAS Output using Base SAS' COLOR = BLUE attr =
underline
#9 @10 'Are we doing Frequency on ALL or select variables?(1-All variables, 2-Select variables)'
COLOR = RED @110 VARS 2 attr=underline color=RED REQUIRED = YES
#14 @10 'Press ENTER to continue' COLOR = RED;
#display start;
%IF &VARS = 2 %THEN %DO;
%window start
#4 @22 'Building Drilldown Capabilities Dynamically in SAS Output using Base SAS' COLOR = BLUE attr =
underline
#9 @10 'Please type in the variable names in single quotes and separated by comma'
COLOR = RED @90 fields 50 attr=underline color=RED REQUIRED = YES
#14 @10 'Press ENTER to continue' COLOR = RED;
%display start;
%END;
%let freq_fields=%UPCASE(&FIELDS);

proc sort data=dsn.&dataset out=&dataset;
by ID;

data All_Freqs;
set _null_;
run;

%macro create_tables(lib, dataset, col,type);
proc freq data=&lib.&dataset;
table &col./list missing out=&col. outcum;
run;

data &col.;
set &col.;
length value $300. varname $20. vartype $4.;
varname="&col.";
uid=_N_;

```



```

value=STRIP(put(&col.,8.));
vartype="&type";
drop &col.;
run;

data All_Freqs;
set All_Freqs &col.;
run;
%mend create_tables;

%macro table_driver;
data _null_;
set sashelp.Vcolumn;
%if &VARS.^=2 %then %do;
if libname='DSN' and memname=upcase("&dataset") and upcase(name) not in ('ID','BATCH');
call execute("%create_tables('||libname||','||memname||','||name||','||type||')");
%end;
%else %do;
/* if only need frequencies on specified variables, uncomment the following statement*/
if libname='DSN' and memname=upcase("&dataset") and upcase(name) in (&freq_fields) ;
call execute("%create_tables('||libname||','||memname||','||name||','||type||')");
%end;
RUN;
%mend table_driver;

%table_driver;

%macro create_htmls(var,value,uid,type);
ODS html file="&path.\html&var._&uid..html";

%if &type=char %then %do;
proc print data=&dataset. noobs;
where &var.="&value.";
var ID Gender Age SchoolID State;
title "Variable: &var., Value:&value.";
run;
%end;
%else %if &type=num %then %do;
%if %sysfunc(anyalpha(&value))>0 %then %do;
/* for value .D, .R, or .M etc */
proc print data=&dataset. noobs;
where &var.=.&value.;
var ID Gender Age SchoolID State;
title "Variable: &var., Value:&value.";
run;
%end;
%else %do;
* pure numbers */
proc print data=&dataset. noobs;
where &var.=&value.;
var ID Gender Age SchoolID State;
title "Variable: &var., Value:&value.";
run;
%end;
%end;

%mend create_htmls;

data _null_;
set All_Freqs;
call execute("%create_htmls('||STRIP(varname)||','||STRIP(value)||','||STRIP(uid)||','||STRIP(vartype)||')");
run;

title "<IMG SRC = 'MPRIlogo.jpg'>";
footnote "<A HREF='http://www.mathematica-mpr.com/'>Mathematica Policy Research Inc.</A>";

```

```

ODS html file="&path.\HTML\freq_&PROJECT..html";

%macro create_report(field,lbl);
proc report data=&field. headskip headline ls=256 nowd missing;
column uid value COUNT PERCENT;
define uid/display noprint;
define value/display "&field." width=255 flow;
define COUNT /'N';
define PERCENT / 'P' f=7.2;
title2 "Frequencies of &field.";

compute COUNT;
urlstring="&path.\HTML\&field._||STRIP(uid)||.html";
if fileexist(urlstring) then do;
call define("_c3_", "URL", urlstring);
CALL DEFINE("_c3_", "style", "STYLE=[BACKGROUND=LIGHT GREEN FONT_WEIGHT = BOLD]");
end;
endcomp;
run;
%mend create_report;

%macro rpt_driver;
/* Frequencies on all variables */
%if &VARS.^=2 %then %do;
data _null_;
set sashelp.Vcolumn;
if libname='DSN' and memname=upcase("&dataset") and upcase(name) not in ('ID','BATCH');
call execute("%create_report(||name||,%str(||label||)");
run;
%END;

/* Frequencies on selected variables */
%ELSE %DO;
data _null_;
set sashelp.Vcolumn;
if libname='DSN' and memname=upcase("&dataset") and upcase(name) in (&freq_fields);
call execute("%create_report(||name||,%str(||label||)");
run;

%END;
%mend rpt_driver;
%rpt_driver;
ODS html close;
%END;

%ELSE %IF &FRQ = 2 %THEN %DO;
/* two way frequencies */
%window start
#4 @22 'Building Drilldown Capabilities Dynamically in SAS Output using Base SAS' COLOR = BLUE attr =
underline
#9 @10 'What would be your row variable in the crosstab?' COLOR = RED @65 var1 15 attr=underline color=RED
REQUIRED = YES
#11 @10 'What would be your column variable in the crosstab?' COLOR = RED @65 var2 15 attr=underline
color=RED REQUIRED = YES
#14 @10 'Press ENTER to continue' COLOR = RED;
%display start;

/* determine the data types of var1 and var2 */
data _null_;
set sashelp.Vcolumn;
if libname=upcase('dsn') and memname=upcase("&dataset.") then do;
if upcase(STRIP(name))=upcase(STRIP("&var1")) then call symput('type1',STRIP(type));
if upcase(STRIP(name))=upcase(STRIP("&var2")) then call symput('type2',STRIP(type));
end;
run;

```

```

/* If value is missing then it will be converted to S */
/* This is to ensure that the title if for missing value will not be blank */
data &dataset;
set dsn.&dataset;
if &type1. NE num or &type2. NE num then do;
array c _char_;
do over c;
if c=' ' then c='S';
end;
array i _numeric_;
do over i;
if i=. then i=.S;
end;
end;
Drop num char;
run;

proc freq data=&dataset noprint;
table &var1.*&var2./ out=outdata missing;
run;

/* Sort by the row field in output */
proc sort data=outdata;
by &var1;
run;

/* Get the unique value by variable */
proc sql;
create table &var2 as
select distinct &var2 from outdata;

create table &var1 as
select distinct &var1 from outdata;
quit;

/* Assign a unique number to each row of data. This number will determine the column number for &var2 value */
/* Create macro variable to determine number of columns */
data &var2;
set &var2 end=l;
col=_N_;
if l=1 then call symput('varnum',STRIP(put(_N_,8.)));
run;

proc sort;
by &var2;

proc sort data=outdata;
by &var2;
data v12c;
merge outdata(in=a keep=&var1 &var2) &var2.(in=b);
by &var2;
run;

proc sort;
by &var1;

/* Assign a unique number to each row of data. This number will determine the row number for &var1 value */
data &var1;
set &var1;
row=_N_;
run;

proc sort;

```

```

by &var1;

/* Merge 2 datasets to get col/row location of each &var1 & &var2 combination */
data v12c_final;
merge v12c(in=a) &var1.(in=b);
by &var1;
run;

proc sort data=&dataset;
by &var1 &var2;

proc sort data=v12c_final;
by &var1 &var2;

/* Merge with main dataset */
data final;
merge &dataset(in=a) v12c_final(in=b);
by &var1 &var2;
if a;
run;

proc sort;
by id;

%macro create_each_html(row,col);

ODS html file="&path.\HTML\&PROJECT._&row._&col..html" STYLE = barrettsblue;
proc print data=FINAL noobs;
where row=&row and col=&col;
VAR ID GENDER AGE SCHOOLID STATE;
title "<IMG SRC = 'MPRIlogo.jpg'>";
title2 "&var1.=&vvar1, &var2.=&vvar2" ;
run;
%mend create_each_html;

data _null_;
set v12c_final;
call symput('vvar1',&var1);
call symput('vvar2',&var2);
call execute('%create_each_html('||STRIP(row)||','||STRIP(col)||');');
run;

proc sort data=outdata;
by &var1;

/* Merge data to get row number of &var1 in cross tab */
data outdata2;
merge outdata(in=a) &var1.(in=b);
by &var1;
if a;
run;

/* Run Proc Report for cross tab */
%macro create;
ODS html file="&path.\HTML\crosstab_&PROJECT..html" STYLE = barrettsblue ;
proc report data=outdata2 headskip headline ls=255 nowd missing;
column row &var1. ("&var2" &var2, (COUNT PERCENT));
define &var1./group "&var1" STYLE=[BACKGROUND= barrettsblue];
define row/group noprint;
define &var2./across ' ' order=INTERNAL;
define COUNT /'N';
define PERCENT / 'P' f=7.2 ;

title "<IMG SRC = 'MPRIlogo.jpg'>";
title2 "Table of &var1 by &var2";

```

```
footnote "<A HREF='http://www.mathematica-mpr.com/'>Mathematica Policy Research Inc.</A>";
compute COUNT;
%Do I=1 %to %eval(&varnum);
urlstring_&I. = "&path.\HTML\&PROJECT._" || STRIP(row) || "_" || STRIP(&I.) || ".html";
if fileexist(urlstring_&I.) and _c%eval(2*&I.+1)_>0 then do;
call define("_c%eval(2*&I.+1)_" , "URL", urlstring_&I. );
CALL DEFINE(_COL_, "style", "STYLE=[FONT_WEIGHT = BOLD]");
end;
%end;
endcomp;
COMPUTE PERCENT;
CALL DEFINE(_COL_, "style", "STYLE=[BACKGROUND=YELLOW]");
ENDCOMP;
run;

%mend CREATE;
%create;
ODS html close;
%END;
%MEND FREQ;
```